Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

Further Connections Between Contract-Scheduling and Ray-Searching Problems*

Spyros Angelopoulos
Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France and
CNRS, UMR 7606, LIP6, F-75005, Paris, France
spyros.angelopoulos@lip6.fr

Abstract

This paper addresses two classes of different, yet interrelated optimization problems. The first class of problems involves a robot that must locate a hidden target in an environment that consists of a set of concurrent rays. The second class pertains to the design of interruptible algorithms by means of a schedule of contract algorithms. We study several variants of these families of problems, such as searching and scheduling with probabilistic considerations, redundancy and fault-tolerance issues, randomized strategies, and trade-offs between performance and preemptions. For many of these problems we present the first known results that apply to multi-ray and multi-problem domains. Our objective is to demonstrate that several well-motivated settings can be addressed using a common approach.

1 Introduction

In this paper we expand the study of connections between two seemingly different, yet interrelated classes of scheduling problems. The first class of problems involves a mobile searcher that must explore an unknown environment so as to locate a hidden target. Objectives of this nature are often encountered in the domain of robotic search and exploration. The second class of problem pertains to the design of a computational multi-problem solver, which may be interrupted at any point in time, and may be queried for its currently best solution to any of the given problems. This setting provides a very practical modeling of situations that often arise in the realm of AI applications, such as the design of any-time and real-time intelligent systems [Zilberstein, 1996].

Searching for a hidden object in an unbounded domain is a fundamental computational problem, with a rich history that dates back to early work [Bellman, 1963] [Beck, 1964] in the context of searching on the infinite line (informally known as the *cow-path problem*). In our work we focus on a generalization of linear search, known as the *star search* or *ray search* problem. Here, we are given a set of m semi-infinite, concurrent rays which intersect at a common origin O, as well as a mobile searcher which is initially placed at the origin. There is also a target that is hidden at some distance d from O, at a ray unknown to the searcher. The objective is to design a search strategy that minimizes the *competitive ratio*, namely the worst-case ratio of the distance traversed by the robot (up to target detection) over the distance d.

Problems related to ray searching have attracted significant interest from the AI/OR communities. Optimal competitive ratios were obtained in [Gal, 1974] and [Baeza-Yates *et al.*, 1993]. The setting in which certain probabilistic information concerning the target placement is known was studied in [Jaillet and Stafford, 1993], [Kao and Littman, 1997]. The effect of randomization on the expected performance was addressed in [Schuierer, 2003], [Kao *et al.*, 1996]. In the case where an upper bound on the distance from the target is known [López-Ortiz and Schuierer, 2001] provides a near-optimal asymptotic analysis, whereas in the case where the searcher incurs a fixed turn cost [Demaine *et al.*, 2006] provides an optimal search strategy. Other work includes the setting of multiple parallel searchers [López-Ortiz and Schuierer, 2004], the related problem of designing *hybrid algorithms* [Kao *et al.*, 1998], and more recently, the study of new performance measures [Kirkpatrick, 2009], [McGregor *et al.*, 2009]. We refer the interested reader to Chapters 8 and 9 of the textbook [Alpern and Gal, 2003] for further results.

The second class of problems is related to bounded-resource reasoning in the context of *anytime* algorithms [Russell and Zilberstein, 1991]. Such algorithms provide a trade-off between computation time and the quality of the output, when there is uncertainty with respect to execution time. More specifically, our goal is to be able to simulate an *interruptible algorithm* by means of repeated executions of a *contract algorithm*. These are both classes of anytime algorithms which, however, differ significantly in terms of their handling of interruptions. On the one hand, an interruptible algorithm will always produce some meaningful result (in accordance to its performance profile) whenever an interruption occurs during its execution. On the other hand, a contract algorithm must be provided, as part of the input, with its allowed, pre-specified computation time (i.e., contract time). If completed by the contract time, the algorithm will always output the solution consistent with its performance profile,

*Research supported by project ANR-11-BS02-0015 "New Techniques in Online Computation–NeTOC".

otherwise it may fail to produce any useful result.

As observed in [Bernstein *et al.*, 2002], contract algorithms tend to be simpler to implement and maintain, however they lack in flexibility compared to interruptible algorithms. This observation raises the challenge of simulating an interruptible algorithm using repeated executions of contract algorithms. The precise framework is as follows: given n instances of optimization problems, and a contract algorithm for each problem, provide a strategy for scheduling repeated executions of a contract algorithm, in either a single, or multiple processors. Upon an interruption, say at time t, the solution to any of the n problems may be requested. The system returns the solution that corresponds to the longest completed execution of a contract algorithm for the problem in question. The standard performance measure of this scheduling strategy is the *acceleration ratio* [Russell and Zilberstein, 1991], which informally implies that an increase of the processor speed by a factor equal to the acceleration ratio of the schedule yields a system which is as efficient as one in which the interruption time is known in advance.

Previous research has established the optimality of scheduling strategies based on iterative deepening methods in the settings of single problem/single processor [Russell and Zilberstein, 1991] [Zilberstein *et al.*, 2003], single problem/multiple processors [Zilberstein *et al.*, 2003] and multiple problems/single processor [Bernstein *et al.*, 2002]. The most general setting of multiple problems and processors was investigated in [Bernstein *et al.*, 2003], which was also the first to demonstrate connections between ray searching and contract scheduling problems. More specifically [Bernstein *et al.*, 2003] shows a reduction between specific classes of search and scheduling strategies known as *cyclic* strategies (see Section 2). Optimal schedules, without restrictions, were established in [López-Ortiz *et al.*, 2014]. Issues related to soft deadlines were addressed in [Angelopoulos *et al.*, 2008], and measures alternative to the acceleration ratio were introduced in [Angelopoulos and López-Ortiz, 2009].

Contribution of this paper In this work we expand the study of connections between the search and scheduling problems that was initiated in [Bernstein *et al.*, 2003]. Namely, we address several settings that provide well-motivated extensions and generalizations of these two classes of problems. More precisely, we study the following problems:

Uncertain target detection / Monte Carlo contract algorithms: We investigate the setting in which the searcher detects the target with probability p during each visit, and the setting in which each contract algorithm is a randomized Monte Carlo algorithm with probability of success equal to p.

Redundancy and fault tolerance: We seek search strategies under the constraint that at least r visits over the target are required in order to locate it. On a similar vain, we seek scheduling strategies under the assumption that at least r executions of a contract algorithm are required so as to benefit from its output. This is related to search and scheduling with uncertainty, when the probability of success is unknown.

Randomized scheduling strategies: We show how access to random bits can improve the expected performance of a scheduling strategy.

Trade-offs between performance and the number of searches and contracts: We quantify the trade-offs between the performance ratios and the number of turns by the searcher or the number of algorithm executions in the schedule.

For all problems, with the exception of randomized strategies, we give the first results (to our knowledge) that apply to both the multi-ray searching and multi-problem scheduling domains. Concerning randomization, we show how to apply and extend, in a non-trivial manner, ideas that stem from known randomized ray-searching algorithms. In addition, we address an open question in [Bernstein *et al.*, 2003], who asked "whether the contract scheduling and robot search problems have similarities beyond those that result from using cyclic strategies". In particular, in Section 4 we present non-cyclic strategies that improve upon the best cyclic ones.

Although we follow an analytical approach in the problems we study, we believe that both the settings and the proposed algorithms are applicable in real-life situations. For instance, scheduling executions of Monte Carlo algorithms can be very useful in algorithm portfolios. Similarly, the trade-offs between performance and the number of executions of searches/contracts can be essential in situations in which the turn/setup cost is very high. As a concrete example, consider drilling for oil in multiple locations (which can be modeled as a ray-searching problem [McGregor *et al.*, 2009]). Here, moving the drill is a very costly operation that needs to be taken into consideration.

Due to space limitations, we omit or sketch certain technical proofs and extensions. We refer the reader to [Angelopoulos, 2015] for the full version of this paper.

2 Preliminaries

Ray searching. We assume a single robot and m rays, numbered $0 \dots m-1$. For a target placement T at distance d from the origin, we define the *competitive ratio* of a strategy as

$$\alpha = \sup_T \frac{\text{cost for locating T}}{d} \quad (1)$$

A strategy is *round-robin* or *cyclic* if it described by an infinite sequence $\{x_i\}_{i=0}^{\infty}$ as follows: in the i-th iteration, the searcher explores ray $(i \bmod m)$ by starting at the origin O, reaching the point at distance x_i from O, and then returning to O. A cyclic strategy is called *monotone*, if the sequence $\{x_i\}_{i=0}^{\infty}$ is non-decreasing. A special class of monotone strategies is the class of *exponential* strategies, namely strategies in which $x_i = b^i$, for some given $b > 1$, which we call the *base* of the strategy. Exponential strategies are often optimal among monotone strategies (see [Alpern and Gal, 2003]), and in many cases they are also globally optimal. Indeed, for m-ray searching, the exponential strategy with base $b = \frac{m}{m-1}$ attains the optimal competitive ratio [Gal, 1972]

$$\alpha^*(m) = 1 + 2\frac{b^m - 1}{b-1},\ b = \frac{m}{m-1}. \quad (2)$$

Note that $\alpha^*(m) = O(m)$, and $\alpha^*(m) \to 1 + 2em$ as $m \to \infty$.

Contract scheduling: We assume a single processor and n problems, numbered $0 \dots n-1$. For interruption time t, let $\ell_{i,t}$ denote the length (duration) of the longest execution of a contract algorithm for problem i that has completed by time t. Then the acceleration ratio of the schedule [Russell and Zilberstein, 1991] is defined as

$$\beta = \sup_{t,i \in [0,\ldots,n-1]} \frac{t}{\ell_{i,t}}. \quad (3)$$

Similar to ray searching, a round-robin or cyclic strategy is described by an infinite sequence $\{x_i\}_{i=0}^{\infty}$ such that in iteration i, the strategy schedules an execution of a contract for problem $(i \bmod n)$, and of length equal to x_i. The definitions of monotone and exponential strategies are as in the context of ray searching, and we note that, once again, exponential strategies often lead to optimal or near-optimal solutions (see, e.g., [Zilberstein *et al.*, 2003] [López-Ortiz *et al.*, 2014], [Angelopoulos *et al.*, 2008]). In particular, for n problems, the exponential strategy with base $b = \frac{n+1}{n}$ attains the optimal acceleration ratio [Zilberstein *et al.*, 2003]

$$\beta^*(n) = \frac{b^{n+1}}{b-1},\ b = \frac{n+1}{n}. \quad (4)$$

Note that $\beta^*(n) = O(n)$, and that $\beta^*(n) \to \mathrm{e}(n+1)$, for $n \to \infty$.

Occasionally, we will make a further distinction between worst-case and asymptotic performance. Namely, the asymptotic competitive ratio is defined as $\lim_{T:d\to\infty} \frac{\text{cost for locating T}}{d}$, whereas the asymptotic acceleration ratio is defined as $\lim_{t\to\infty} \sup_{i\in[0,\ldots,n-1]} \frac{t}{l_{i,t}}$ (assuming that the measures converge to a limit).

3 Search with probabilistic detection and scheduling of randomized contracts

In this section we study the effect of uncertainty in search and scheduling. In particular, we consider the setting in which the detection of a target is stochastic, in that the target is revealed with probability p every time the searcher passes over it. Similarly, we address the problem of scheduling randomized contract algorithms; namely, each execution of the (Monte Carlo) randomized algorithm succeeds with probability p. This variant has been studied in [Alpern and Gal, 2003] only in the context of linear search (i.e., when $m = 2$), and the exact competitiveness of the problem is not known even in this much simpler case. No results are known for general m.

In this setting, the search cost is defined as the expected time of the first successful target detection. Moreover, for every problem i and interruption t, we define $\mathbb{E}[\ell_{i,t}]$ as the expected longest contract completed for problem i by time t. The competitive and the acceleration ratios are then defined naturally as extensions of (1) and (3).

We begin with a lower bound on our measures.

Lemma 1. *Every search strategy with probabilistic detection has competitive ratio at least $\frac{m}{2p}$, and every scheduling strategy of randomized contract algorithms has acceleration ratio at least $\frac{n}{p}$.*

Proof. Consider first the search variant. Let S denote the set of all points at distance at most d from the origin. Given a point $x \in S$, let t_x^k denote the time in which the searcher reaches x for the k-th time. It is not difficult to show that for every $k \geq 1$, there exists $x \in S$ such that the search cost at the time of the k-th visit of x is at least $kmd/2$. This implies that targets in S are detected at expected cost at least $\sum_{k=1}^{\infty} p(1-p)^{k-1} t_x^k \geq \sum_{k=1}^{\infty} p(1-p)^{k-1} kmd/2 = md/(2p)$; here, we used the expectation of the geometric distribution. The result follows from (1).

Consider now the scheduling variant. For a given interruption time t and a given problem instance i, let $l_1^i, l_2^i, \ldots l_{n_i}^i$ denote the lengths of the contracts for problem i that have completed by time t, in non-increasing order. Let the random variable $\ell_{i,t}$ denote the expected length of the longest contract completed for problem i by time t. Then $\mathbb{E}[\ell_{i,t}] = \sum_{j=1}^{n_i} p(1-p)^{j-1} l_j^i \leq p \sum_{j=1}^{n_i} l_j^i$. Since $\sum_{i=0}^{n-1} \sum_{j=1}^{n_i} l_j^i = t$, there exists a problem i for which $\mathbb{E}[\ell_{i,t}] \leq p\frac{t}{n}$. The claim follows from the definition of acceleration ratio (3). ☐

Theorem 2. *There exists an exponential strategy for searching with probabilistic detection that has competitive ratio at most $1 + 8\frac{m}{p^2}$.*

Proof. Let $\{x_i\}_{i=0}^{\infty}$ denote the searcher's exponential strategy, where $x_i = b^i$, for some b that will be chosen later in the proof. Let d denote the distance of the target from the origin, then there exists index l such that $x_l < d \leq x_{l+m}$. We denote by P_k the probability that the target is found during the k-th visit of the searcher, when all previous $k-1$ attempts were unsuccessful, hence $P_k = (1-p)^{k-1}p$. We also define q_j as the probability that the target is found after at least j attempts, therefore we have $q_j \doteq \sum_{k=j}^{\infty} P_k = (1-p)^{j-1}$.

In order to simplify the analysis, we will make the assumption that the searcher can locate the target only while it is moving away from the origin (and never while moving towards the origin); it turns out that this assumption weakens the result only by a constant multiplicative factor.

We first derive an expression for the expected cost C of the strategy. Note that first time the searcher passes the target, it has traveled a total distance of at most $2\sum_{j=0}^{l+m-1} x_j + d$; more generally, the total distance traversed by the searcher at its k-th visit over the target is at most $2\sum_{j=0}^{l+km-1} x_j + d$. We obtain that the expected cost is bounded by

$$C = \sum_{k=1}^{\infty} P_k (2 \sum_{j=0}^{l+mk-1} x_j + d), \quad (5)$$

from which we further derive (using the connection between P_k and q_j) that the competitive ratio of the strategy is

$$\alpha \leq \frac{C}{d} \leq 1 + \frac{2}{x_l} \sum_{k=1}^{\infty} P_k \sum_{j=0}^{l+mk-1} x_j$$
$$= 1 + \frac{2}{x_l} \sum_{j=0}^{l} x_{j+m-1} + \frac{2}{x_l} \sum_{j=2}^{\infty} q_j \sum_{i=1}^{m-1} x_{l+(j-1)m+i} \quad (6)$$

By rearranging the terms in the summations we observe that

$$\sum_{j=2}^{\infty} q_j \sum_{i=1}^{m-1} x_{l+(j-1)m+i} = \sum_{i=1}^{m-1} \sum_{j=2}^{\infty} q_j x_{l+(j-1)m+i}$$
$$= \sum_{i=1}^{m-1} x_{l+i} \sum_{j=2}^{\infty} ((b^m(1-p))^{j-1}. \quad (7)$$

By defining $\lambda \doteq b^m(1-p)$, and by combining inequalities (6) and (7) we obtain that the competitive ratio is at most $\alpha \leq 1 + 2\frac{b^m}{b-1}\sum_{j=0}^{\infty} \lambda^j$. Note that unless $\lambda < 1$ the competitive ratio is not bounded. Assuming that we can choose $b > 1$ such that $\lambda < 1$, the competitive ratio is

$$\alpha \leq 1 + 2\frac{b^m}{b-1} \cdot \frac{1}{1-\lambda}. \quad (8)$$

We will show how to choose the appropriate $b > 1$ so as to guarantee the desired competitive ratio. To this end, we will first need the following technical lemma.

Lemma 3. *The function $f : \mathbb{R}^+ \to \mathbb{R}$ with $f(x) = \mathrm{e}^x(1-p) + \mathrm{e}^x \frac{p^2}{4x} - 1$ has a root r such that $0 < r \leq \frac{p}{2}$.*

Let r denote the root of the function f, defined in the statement of Lemma 3. We will show that choosing base $b = \frac{m}{m-r}$ yields the desired competitive ratio. It is straightforward to verify that $b > 1$ and that $\lambda = b^m(1-p) < 1$. Hence, the competitive ratio converges to the value given by the RHS of (8). From the choice of b, we have that $b - 1 = \frac{r}{m-r}$ and $b^m \leq e^r$. We then obtain $\frac{b^m}{b-1} \cdot \frac{1}{1-\lambda} \leq \frac{e^r(m-r)}{r\left((1-(1-p)\left(\frac{m}{m-r}\right)^m\right)} \leq \frac{me^r}{r((1-(1-p)\mathrm{e}^r)}$. Recall that from Lemma 3, r is such that $1-(1-p)\mathrm{e}^r = \mathrm{e}^r \frac{p^2}{4}$. We thus obtain that $\frac{b^m}{b-1}\frac{1}{1-\lambda} \leq \frac{m}{4p^2}$, and from (8) it follows that the competitive ratio of the strategy is at most $1 + 8m/p^2$. ☐

Theorem 4. *There exists an exponential strategy for scheduling randomized contract algorithms that has acceleration ratio at most $\mathrm{e}\frac{n}{p} + \frac{\mathrm{e}}{p}$.*

Proof. Let b denote the base of the exponential strategy. It is easy to see that the acceleration ratio is maximized for interruptions t that are arbitrarily close to, but do not exceed the finish time of a contract. Let t denote such an interruption time, in particular right before termination of contract $i + n$, for some $i > 0$; in other words, $t = \frac{b^{i+n+1}-1}{b-1}$. Then every problem has completed a contract of expected length at least pb^i by time t. Therefore, the acceleration ratio of the schedule is at most $\beta \leq \sup_{i>0} \frac{b^{n+i+1}}{pb^i(b-1)}$, and choosing $b = \frac{n+1}{n}$ we obtain that $\beta \leq \mathrm{e}\frac{n}{p} + \frac{\mathrm{e}}{p}$, since $(1 + 1/n)^n \leq \mathrm{e}$. ☐

4 Fault tolerance/redundancy in search and scheduling

In Section 3 we studied the searching and scheduling problems in a stochastic setting. But what if the success probability is not known in advance? In the absence of such information, one could opt for imposing a lower bound r on the number of times the searcher has to visit the target and, likewise, a lower bound r on the number of times a contract algorithm must be executed before its response can be trusted. Alternatively, this setting addresses the issues of fault tolerance and redundancy in the search and scheduling domains. The search variant has been studied in [Alpern and Gal, 2003] only in the context of linear search ($m = 2$); as in the case of probabilistic detection, even when $m = 2$ the exact optimal competitive strategies are not known.

The following lemma follows using an approach very similar to the proof of Lemma 1.

Lemma 5. *Every search strategy on m rays with redundancy guarantee $r \in \mathbb{N}^+$ has competitive ratio at least $\frac{rm}{2}$.*

We first evaluate the best exponential strategy. This can be done using standard techniques (see e.g., [Bernstein *et al.*, 2003]), and thus we omit the proof of the following theorem.

Theorem 6. *The best exponential strategy has competitive ratio $2(\lceil\frac{r}{2}\rceil m - 1)\left(\frac{\lceil\frac{r}{2}\rceil m}{\lceil\frac{r}{2}\rceil m-1}\right)^{\lceil\frac{r}{2}\rceil m} + 1$, if r is odd, and $2(\lceil\frac{r}{2}\rceil m - 1)\left(\frac{\lceil\frac{r}{2}\rceil m}{\lceil\frac{r}{2}\rceil m-1}\right)^{\lceil\frac{r}{2}\rceil m} - 1$, if r is even.*

Theorem 6 implies that the best exponential strategy has competitive ratio at most $2\mathrm{e}(\lceil\frac{r}{2}\rceil(m-1)) \pm 1$ (depending on whether r is even or odd). Interestingly, we can show that there exist non-monotone strategies, which, for $r > 2$, improve upon the (best) exponential strategy of Theorem 6. For simplicity, let us assume that r is even, although the same approach applies when r is odd. Consider the following strategy: In iteration i, the searcher visits ray $i \bmod m$ first up to the point at distance x_{i-m}, then performs r traversals of the interval $[x_{i-m}, x_i]$ (thus visiting r times each point of the said interval), then completes the iteration by returning to the origin (here we define $x_j = 0$ for all $j < 0$). We call this strategy NM-SEARCH (non-monotone search).

Theorem 7. *Strategy NM-SEARCH has competitive ratio at most $r(m-1)\left(\frac{m}{m-1}\right)^m + 2 - r$.*

Proof. Suppose that the target lies at a distance d from the origin, and let $l \in N$ denote an index such that $x_l < d \leq x_{l+m}$. Then the cost of locating the target is at most

$$\sum_{j=0}^{l+m}(r(x_j - x_{j-m}) + 2x_{j-m}) = r \cdot \sum_{j=0}^{m+l} x_j + (2-r)\sum_{j=0}^{l} x_j.$$

Setting $x_i = b^i$ (which we will fix shortly), and given that $d > x_l$, we obtain that the competitive ratio is at most

$$\alpha \leq rb\frac{b^{m+1}}{b-1} + (2-r)\frac{b^l - 1}{b^l(b-1)} \leq \frac{rb^{m+1}}{b-1} + (2-r), \quad (9)$$

where the last inequality follows from the fact that $b^l > 1$. We now observe that (9) is minimized for $b = \frac{m}{m-1}$. Substituting in (9) yields $\alpha \leq r(m-1)\left(\frac{m}{m-1}\right)^m + 2 - r$. ☐

By comparing Theorems 6 and 7, we deduce that the non-monotone strategy is superior to the best exponential strategy for $r > 2$.

Consider now contract scheduling with redundancy parameter r, in the sense that the interruptible system may output only the solutions of contracts that have been executed at least r times by time t. In this setting, the best schedule is derived from a pseudo-exponential strategy, which is defined in phases as follows: in phase $i \geq 0$, r contracts for problem $i \mod n$, and of length b^i are executed, for given base $b > i$. It turns out that this strategy attains the optimal acceleration ratio. The proof of the following theorem uses techniques from [López-Ortiz *et al.*, 2014].

Theorem 8. *The pseudo-exponential scheduling strategy with base* $b = \frac{n+1}{n}$ *has acceleration ratio at most* $rn\left(\frac{n+1}{n}\right)^{n+1}$. *Furthermore, this is optimal.*

A different setting stipulates that the schedule returns, upon interruption t and for queried problem p, the r-th smallest contract for problem p that has completed its execution by time t. In this setting, we can still apply the pseudo-exponential strategy (which is clearly non-monotone). We can show, as in ray searching, that this strategy is better than the best exponential strategy, albeit slightly so.

The strategies described above establish connections beyond those that result from the use of cyclic strategies. More precisely, we have shown that non-cyclic ray-searching algorithms have counterparts in the domain of contract-scheduling; furthermore, the non-cyclic strategies improve upon the best cyclic ones. We have thus addressed an open question from [Bernstein *et al.*, 2003], who asked whether there exist connections between the two problems that transcend cyclicality.

5 Randomized scheduling of contract algorithms

In this section we study the power of randomization for scheduling (deterministic) contract algorithms. Our approach is motivated by the randomized strategy of [Kao *et al.*, 1996] for searching on m rays. We emphasize, however, that our analysis differs in several key points, and most notably on the definition of appropriate random events.

We will analyze the following strategy : We choose a random permutation π of the n problems, as well as a random ε uniformly distributed in $[0, 1)$. In iteration $i \geq 0$, the algorithm executes a contract for problem $\pi(i) \mod n$ of length $b^{1+\varepsilon}$, with $b > 1$. This is essentially the algorithm of [Kao *et al.*, 1996], cast in the domain of scheduling (note that in [Kao *et al.*, 1996] ray $\pi(i)$ is searched up to distance $b^{1+\varepsilon}$).

Theorem 9. *The acceleration ratio of the randomized strategy is* $\beta_r(n, b) \leq n\frac{b^{n+1}\ln b}{(b^n-1)(b-1)}$.

Proof. Let t denote the interruption time. Observe that t can be expressed as $t = \frac{b^k-1}{b-1}b^\delta$, for some unique $k \in \mathbb{N}$ and δ such that $1 \leq b^\delta < \frac{b^{k+1}-1}{b^k-1}$. For convenience, we will call the contract execution of length $b^{i+\varepsilon}$ the *i-th contract* of the strategy, and i the contract *index* (with $i \geq 0$). Note that the start and finish times of the i-th contract are $\frac{b^i-1}{b-1}b^\varepsilon$ and $\frac{b^{i+1}-1}{b-1}b^\varepsilon$, respectively.

First, we need to identify the index of the contract during the execution of which the interruption time t occurs; denote this index by l. Note that it cannot be that $l \geq k+1$, since $\frac{b^{k+1}-1}{b-1}b^\varepsilon \geq \frac{b^{k+1}-1}{b-1} > t$. Similarly, it cannot be that $l \leq k-2$ because $\frac{b^{k-1}-1}{b-1}b^\varepsilon \leq \frac{b^k-b}{b-1} < \frac{b^k-1}{b-1} \leq t$. We conclude that either $l = k$, or $l = k-1$. In particular, the random event $(l = k-1)$ occurs only when $\frac{b^k-1}{b-1}b^\varepsilon \geq t = \frac{b^k-1}{b-1}b^\delta$, which implies that $\varepsilon \geq \delta$.

Next, we need to evaluate the expected value of the random variable D that corresponds to the length of the longest contract for the problem that is requested at time t, and which has completed at time t. This will allow us to bound the acceleration ratio α of the randomized strategy, as

$$\sup_t \frac{t}{\mathbb{E}[D]}, \text{ with } t = \frac{b^k-1}{b-1}b^\delta \leq \frac{b^{k+1}-1}{b-1}. \quad (10)$$

We consider two cases, depending on whether $\delta \geq 1$.
Case 1: $\delta \geq 1$. In this case, $\varepsilon < \delta$, which implies, from the above discussion that $k = l$. Therefore, the strategy will return one of the contracts with indices $k-1, k-2, \ldots, k-n$, namely the contract that corresponds to the requested problem. Due to the random permutation of problems performed by the strategy, each of these indices is equally probable to correspond to the requested problem. We thus obtain $\mathbb{E}[D] = \mathbb{E}[D \mid (k = l)] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}[b^{k-i+\varepsilon}] = \frac{1}{n}\sum_{i=1}^n b^{k-i}\frac{b-1}{\ln b} = \frac{1}{n}\frac{b^k(b^n-1)}{b^n \ln b}$, where we used the fact that ε is uniformly distributed in $[0, 1)$. Combining with (10) we obtain $\beta_r(n, b) \leq \frac{b^{k+1}-1}{b^-1}\frac{1}{\mathrm{E}[D]} \leq n\frac{b^{n+1}\ln b}{(b^n-1)(b-1)}$.
Case 2: $0 \leq \delta < 1$, in other words, $b^\delta < b$. Note that in this case, the events $(l = k)$ and $(\varepsilon < \delta)$ are equivalent; similarly for the events $(l = k-1)$ and $(\varepsilon \geq \delta)$. The following technical lemma establishes $\mathbb{E}[D]$ in this case.

Lemma 10. $\mathbb{E}[D] = \frac{1}{n}\frac{b^{k-1}(b^n-1)b^\delta}{b^n}\ln b$.

Combining Lemma 10 and (10) we obtain again that $\beta_r(n, b) \leq \frac{(b^k-1)b^\delta}{b^-1}\frac{1}{\mathrm{E}[D]} \leq n\frac{b^{n+1}\ln b}{(b^n-1)(b-1)}$. □

5.1 Evaluation of the randomized strategy

In order to evaluate the best randomized exponential strategy, we must find the b that minimizes the function $\beta_r(n, b) = n\frac{b^{n+1}\ln b}{(b^n-1)(b-1)}$ (we assume, without loss of generality, that the bound of Theorem 9 may hold with equality). It is easy to see, using standard calculus, that $\beta_r(n, b)$ has a unique minimum, for given n. However, there is no closed form for $\beta_r^*(n) = \min_{b>1}\beta_r(n, b)$. Thus, we must resort to numerical methods.

Figure 1 illustrates the performance of the randomized strategy $\beta_r^*(n)$ versus the deterministic optimal strategy, denoted by $\beta^*(n)$. We observe that $\beta_r^*(n) \leq 0.6\beta^*(n)$, for $n = 1, \ldots 80$. In fact, we can show analytically that for $n \to \infty$, $\beta_r^*(n)$ converges to a value that does not exceed $\frac{\mathrm{e}}{\mathrm{e}-1}(n+1)$ (recall that $\beta^*(n)$ converges to $\mathrm{e}(n+1)$). More precisely, choosing $b = \frac{n+1}{n}$ we obtain $\beta_r^*(n) \leq (n+1)\frac{(1+1/n)^n\ln(1+1/n)}{((1+1/n)^n-1)(1+1/n)}$, which converges to $(n+1)\frac{\mathrm{e}}{\mathrm{e}-1}$, a value extremely close to the computational results.

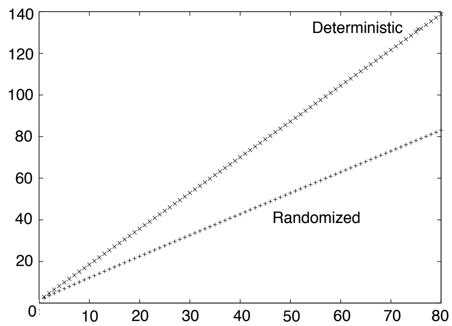

Figure 1: Plots of the randomized ($\beta_r^*(n)$) and the deterministic ($\beta^*(n)$) acceleration ratios, as functions of n.

6 Trade-offs between performance and executions of searches/algorithms

Most previous work on ray searching assumes that the searcher can switch directions at no cost. In practice, turning is a costly operation in robotics, and thus should not be ignored. In a similar vein, we usually assume that there is no setup cost upon execution of a contract algorithm, however some initialization cost may be incurred in practice. One could address this requirement by incorporating the turn/setup cost in the performance evaluation (see [Demaine *et al.*, 2006] for ray searching with turn cost). In this section we follow a different approach by studying the trade-off between performance and the number of searches and/or executions of algorithms.

We will make a distinction between two possible settings. In the first setting, we use the standard definitions of search and scheduling as given in Section 1. Specifically, we address the question: Given a target at distance t (resp. an interruption t) what is the minimum number of turns (resp. executions of contracts) so as to guarantee a certain competitive ratio (resp. acceleration ratio)? We call this the *standard* model.

The second setting is motivated by applications in which searching previously explored territory comes at no cost. One such example is the expanding search paradigm [Alpern and Lidbetter, 2013]. Another example is parallel linear searching on arrays modeled as ray searching [Kirkpatrick, 2009], in which the searcher can "jump" to the last-explored position.

While the latter setting does not have a true counterpart in the realm of contract scheduling, it still gives rise to a scheduling problem. Suppose we have n problems, each with its own statement of an *interruptible* algorithm (as opposed to a contract algorithm). In addition, we allow the use of *preemptions*, in that we can preempt, and later resume the execution of an algorithm. In this context, we face the scheduling problem of interleaving the executions of interruptible algorithms. Note that we can still use the acceleration ratio, given by (3)) as the performance measure, with the notable difference that here $\ell_{i,t}$ denotes the *total* (aggregate) time of algorithm executions for problem i, by time t. We call the above model the *preemptive* model.

Due to space limitations, we only present results on the scheduling problems. We emphasize that similar results can be obtained for ray searching, with only minor modifications.

6.1 Trade offs in the preemptive model

We consider first the problem of scheduling interleaved executions of interruptible algorithms. Clearly, the optimal acceleration ratio is n: simply assign each time unit uniformly across all problems, in a round-robin fashion. However, this optimal strategy results in a linear number of preemptions, as function of time. We thus consider the following *geometric* round-robin strategy, which is a combination of uniform and exponential strategies. The strategy works in phases; namely, in phase i ($i \geq 0$), it executes algorithms for problems $0 \ldots n-1$ with each algorithm allotted a time span equal to b^i, for fixed $b > 1$ (we will call each algorithm execution for problem i a *job* for problem i).

Lemma 11. *The geometric strategy has (worst-case) acceleration ratio $n(b+1)$, asymptotic acceleration ratio nb, and for any t, the number of preemptions incurred up to t is at most $n \log_b \left(\frac{t(b-1)}{n} + 1\right) + n$.*

We will now show that the geometric strategy attains essentially the optimal trade-offs.

Theorem 12. *For any strategy with (worst-case) acceleration ratio $n(1+b) - \epsilon$ for any $b > 1$, and constant $\epsilon > 0$, there exists t such that the number of preemptions up to time t is at least $n \log_b \left(\frac{t(b-1)}{n} + 1\right) - n$. Moreover, any strategy with asymptotic acceleration ratio $nb(1-\epsilon)$, for any constant $\epsilon > 0$, incurs $n \log_b \left(\frac{t(b-1)}{n} + 1\right) - o\left(n \log_b \left(\frac{t(b-1)}{n} + 1\right)\right)$ preemptions by time $t > t_0$, for some t_0.*

Proof. First, suppose, that a strategy S has (worst-case) acceleration ratio $\beta = n(b+1) - \epsilon$, and incurs fewer than $n \log_b \left(\frac{t(b-1)}{n} + 1\right) - n$ preemptions for any t. We can show that there exists a strategy S' of acceleration ratio at least $b(n+1)$, by considering at interruption at time $t = b(n+1)$. Moreover, we can show that S' has at least as good an acceleration ratio as S, which is a contradiction.

For the second part of the theorem, fix a strategy S of asymptotic acceleration ratio $\beta = nb(1-\epsilon)$. Consider a partition of the timeline in phases, such that the i-th phase ($i \geq 0$) spans the interval $[n \sum_{j=0}^{i-1} b^j, n \sum_{j=0}^{i} b^j)$, and thus has length nb^i. We will show that there exists $i_0 > 0$ such that for all $i \geq i_0$, S must incur at least n preemptions in its i-th phase. Since the geometric strategy with base b incurs exactly n preemptions in this interval, for all i, this will imply that we can partition the timeline $t \geq i_0$ in intervals with the property that in each interval, S incurs at least as many preemptions as the geometric strategy, which suffices to prove the result.

Suppose, by way of contradiction, that S incurred at most $n-1$ preemptions within $T = [n \sum_{j=0}^{i-1} b^j, n \sum_{j=0}^{i} b^j]$. Therefore, there exists at least one problem p with no execution in T. Consider an interruption at time $t = n \sum_{j=0}^{i} b^j - \delta$, for arbitrarily small $\delta > 0$. Thus, the aggregate job length for p by time t in S is $\ell_{p,t} \leq n \sum_{j=0}^{i-1} b^j = n \frac{b^i - 1}{b-1}$. Since S

has asymptotic acceleration ratio β, there must exist i_0 and ϵ' with $0 < \epsilon' < \epsilon$ such that for all $i \geq i_0$, $n\frac{b^{i+1}-1}{b-1} - \delta \leq nb(1-\epsilon')\frac{b^i-1}{b-1}$, which it turn implies that $\epsilon'\frac{b^i-b}{b-1} \leq \delta$ for all $i > i_0$. This is a contradiction, since ϵ' depends only on i_0, and δ can be arbitrarily small. □

6.2 Trade offs in the standard model

The ideas of Section 6.1 can also be applied in the standard model. In this setting, however, exponential strategies are a more suitable candidate.

Theorem 13. *For contract scheduling, the exponential strategy with base* b *has acceleration ratio* $\frac{b^{n+1}}{b-1}$*, and schedules at most* $\log_b(t(b-1)+1)+1$ *contracts by* t*. Moreover, any strategy with acceleration ratio at most* $\frac{b^{n+1}}{b-1} - \epsilon$ *for* $b > 1$*, and any* $\epsilon > 0$ *must schedule at least* $\log_b(t(b-1)+1) - o(\log_b(t(b-1)+1))$ *contracts by* t*, for all* $t \geq t_0$.

7 Conclusion

In this paper we demonstrated that many variants of searching for a target on concurrent rays and scheduling contract algorithms on a single processor are amenable to a common approach. There are some intriguing questions that remain open. Can we obtain a $\Theta(m/p)$-competitive algorithm for searching with probabilistic detection? We believe that cyclic strategies are not better than $\Theta(m/p^2)$-competitive. What are the optimal (non-monotone) algorithms for searching/scheduling with redundancy? Note that the precise competitive ratio of these problems is open even when $m = 2$. As a broader research direction, it would be very interesting to address searching and scheduling in heterogeneous environments. For example, one may consider the setting in which each ray is characterized by its own probability of successful target detection.

References

[Alpern and Gal, 2003] S. Alpern and S. Gal. *The theory of search games and rendezvous*. Kluwer Academic Publishers, 2003.

[Alpern and Lidbetter, 2013] S. Alpern and T. Lidbetter. Mining coal or finding terrorists: The expanding search paradigm. *Operations Research*, 61(2):265–279, 2013.

[Angelopoulos and López-Ortiz, 2009] S. Angelopoulos and A. López-Ortiz. Interruptible algorithms for multi-problem solving. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 380–386, 2009.

[Angelopoulos *et al.*, 2008] S. Angelopoulos, A. López-Ortiz, and A. Hamel. Optimal scheduling of contract algorithms with soft deadlines. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI)*, pages 868–873, 2008.

[Angelopoulos, 2015] S. Angelopoulos. Further connections between contract-scheduling and ray-searching problems. *arxiv:1504.07168 [cs:AI]*, 2015.

[Baeza-Yates *et al.*, 1993] R. Baeza-Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Information and Cmputation*, 106:234–244, 1993.

[Beck, 1964] A. Beck. On the linear search problem. *Naval Research Logistics*, 2:221–228, 1964.

[Bellman, 1963] R. Bellman. An optimal search problem. *SIAM Review*, 5:274, 1963.

[Bernstein *et al.*, 2002] D.S. Bernstein, T. J. Perkins, S. Zilberstein, and L. Finkelstein. Scheduling contract algorithms on multiple processors. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI)*, pages 702–706, 2002.

[Bernstein *et al.*, 2003] D.S. Bernstein, L. Finkelstein, and S. Zilberstein. Contract algorithms and robots on rays: unifying two scheduling problems. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1211–1217, 2003.

[Demaine *et al.*, 2006] E.D. Demaine, S.P. Fekete, and S. Gal. Online searching with turn cost. *Theoretical Computer Science*, 361:342–355, 2006.

[Gal, 1972] S. Gal. A general search game. *Israel Journal of Mathematics*, 12:32–45, 1972.

[Gal, 1974] S. Gal. Minimax solutions for linear search problems. *SIAM J. on Applied Math.*, 27:17–30, 1974.

[Jaillet and Stafford, 1993] P. Jaillet and M. Stafford. Online searching. *Operations Research*, 49:234–244, 1993.

[Kao and Littman, 1997] M-Y. Kao and M.L. Littman. Algorithms for informed cows. In *Proceedings of the AAAI 1997 Workshop on Online Search*, 1997.

[Kao *et al.*, 1996] M-Y. Kao, J.H. Reif, and S.R. Tate. Searching in an unknown environment:an optimal randomized algorithm for the cow-path problem. *Inform. and Comp.*, 131(1):63–80, 1996.

[Kao *et al.*, 1998] M-Y. Kao, Y. Ma, M. Sipser, and Y.L. Yin. Optimal constructions of hybrid algorithms. *Journal of Algorithms*, 29(1):142–164, 1998.

[Kirkpatrick, 2009] D. G. Kirkpatrick. Hyperbolic dovetailing. In *Proceedings of the 17th Annual European Symposium on Algorithms (ESA)*, pages 616–627, 2009.

[López-Ortiz and Schuierer, 2001] A. López-Ortiz and S. Schuierer. The ultimate strategy to search on m rays. *Theoretical Computer Science*, 261(2):267–295, 2001.

[López-Ortiz and Schuierer, 2004] A. López-Ortiz and S. Schuierer. On-line parallel heuristics, processor scheduling and robot searching under the competitive framework. *Theor. Comp. Sci.*, 310(1–3):527–537, 2004.

[López-Ortiz *et al.*, 2014] A. López-Ortiz, S. Angelopoulos, and A.M. Hamel. Optimal scheduling of contract algorithms for anytime problems. *Journal of Artificial Intelligence Research*, (51):533–554, 2014.

[McGregor *et al.*, 2009] A. McGregor, K. Onak, and R. Panigrahy. The oil searching problem. In *Proc. of the 17th European Symposium on Algorithms (ESA)*, pages 504–515, 2009.

[Russell and Zilberstein, 1991] S. J. Russell and S. Zilberstein. Composing real-time systems. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 212–217, 1991.

[Schuierer, 2003] S. Schuierer. A lower bound for randomized searching on m rays. In *Computer Science in Perspective*, pages 264–277, 2003.

[Zilberstein *et al.*, 2003] S. Zilberstein, F. Charpillet, and P. Chassaing. Real-time problem-solving with contract algorithms. *Annals of Mathematics and Artificial Intelligence*, 39(1–2):1–18, 2003.

[Zilberstein, 1996] S. Zilberstein. Using anytime algorithms in intelligent systems. *AI Mag.*, 17(3):73–83, 1996.